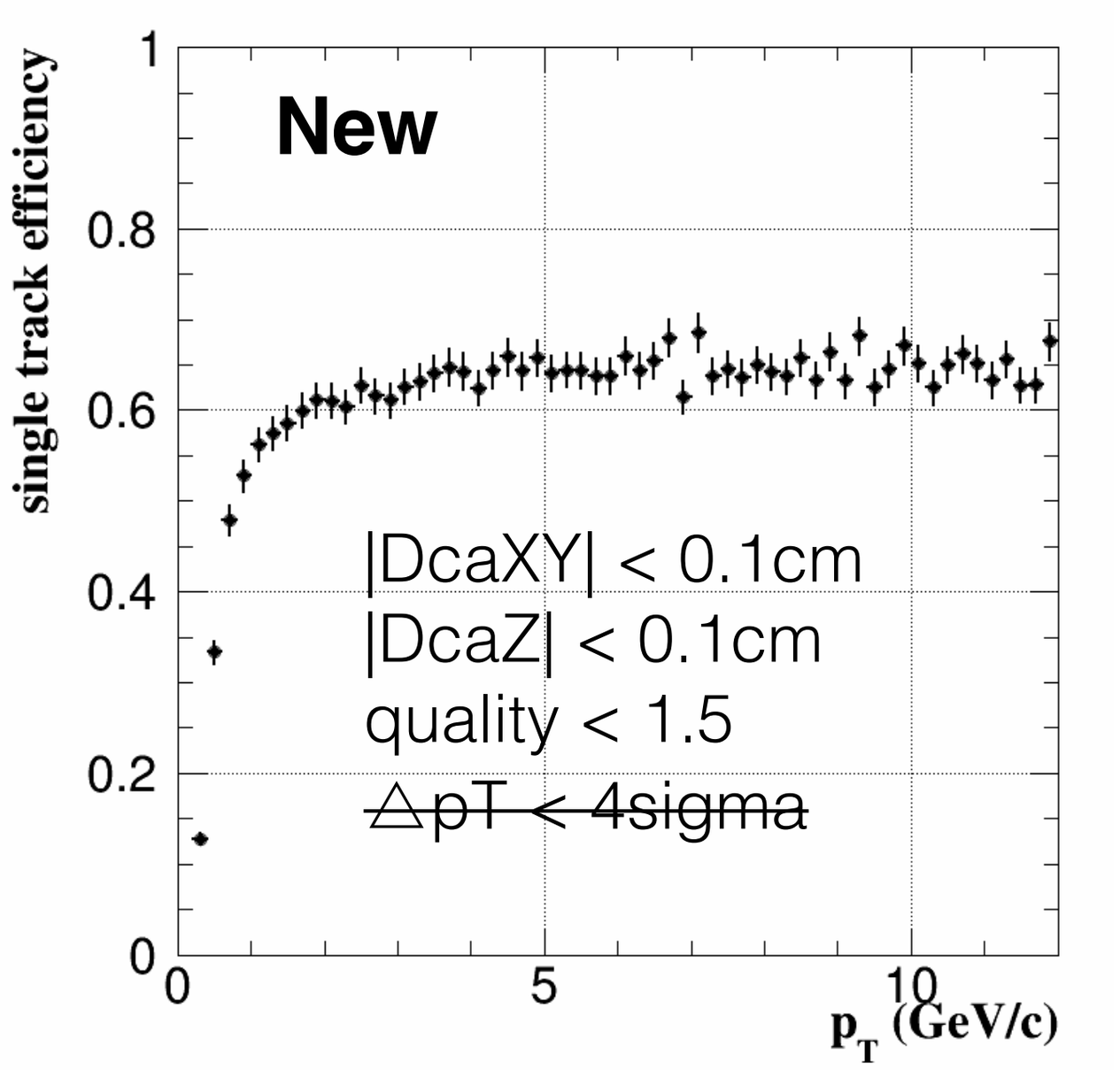
Full Geant4 simulation test

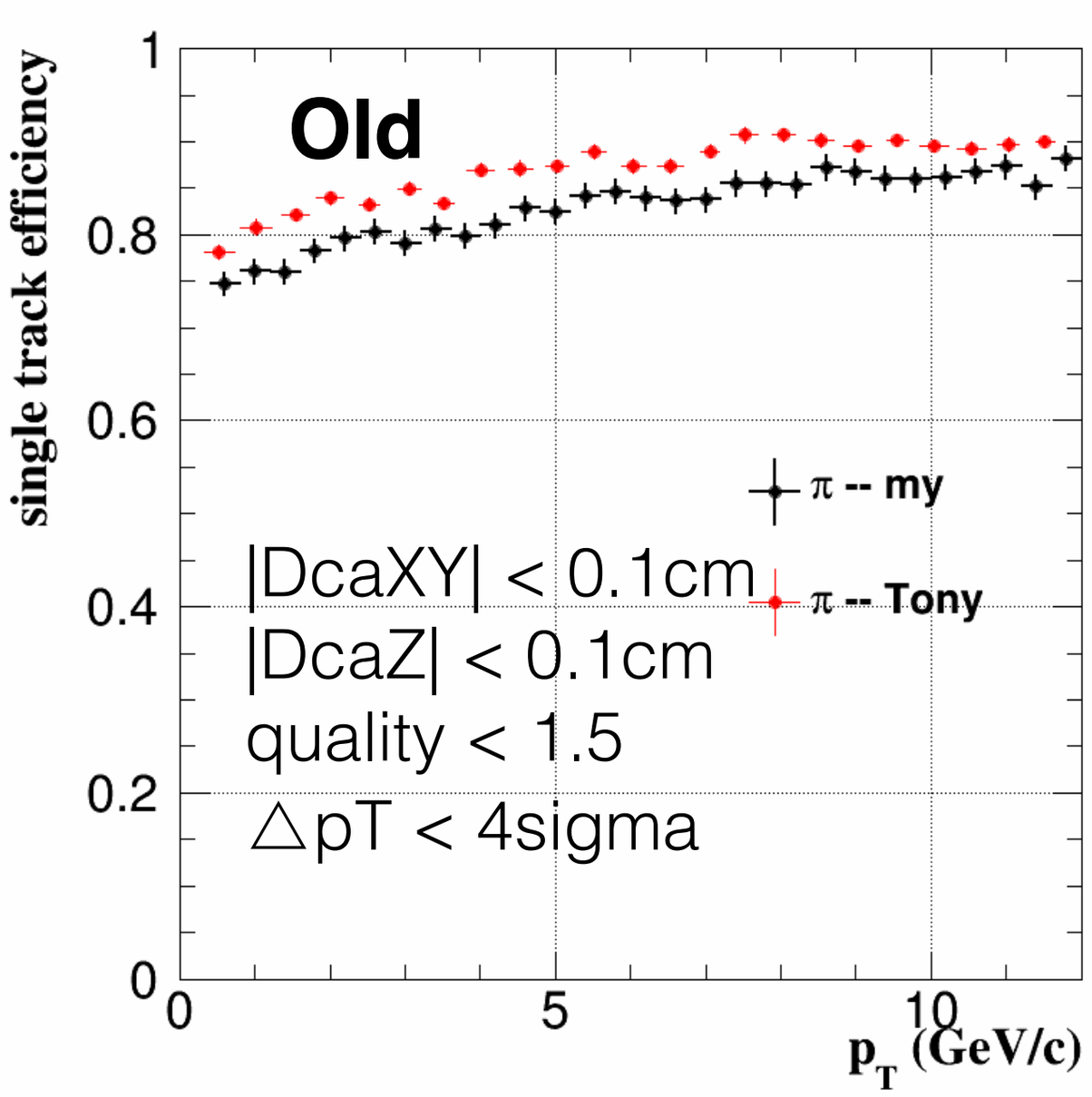
central Hijing + 100 pi/k/p

2017-06-21

Xiaolong Chen

Track eff.

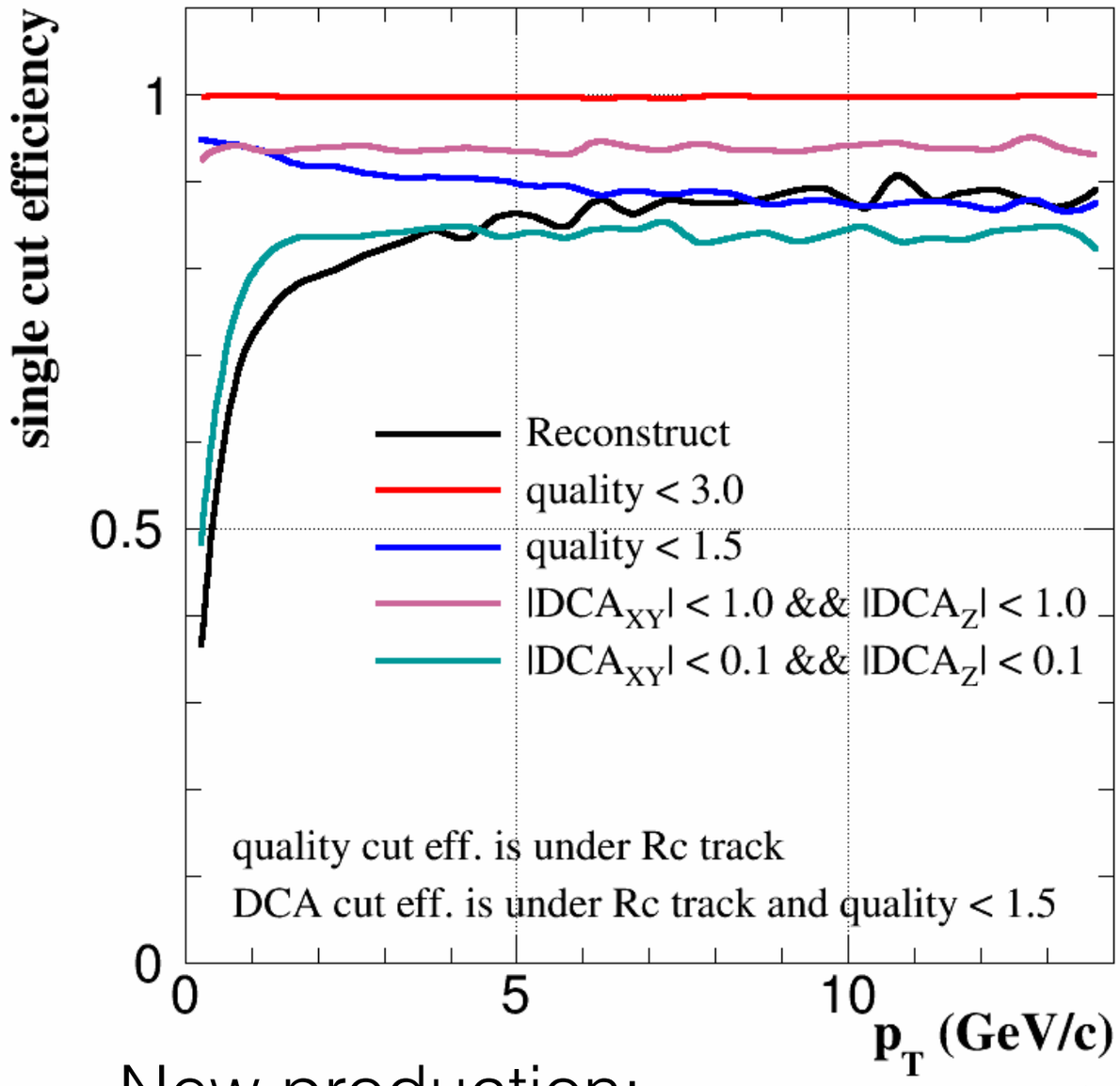




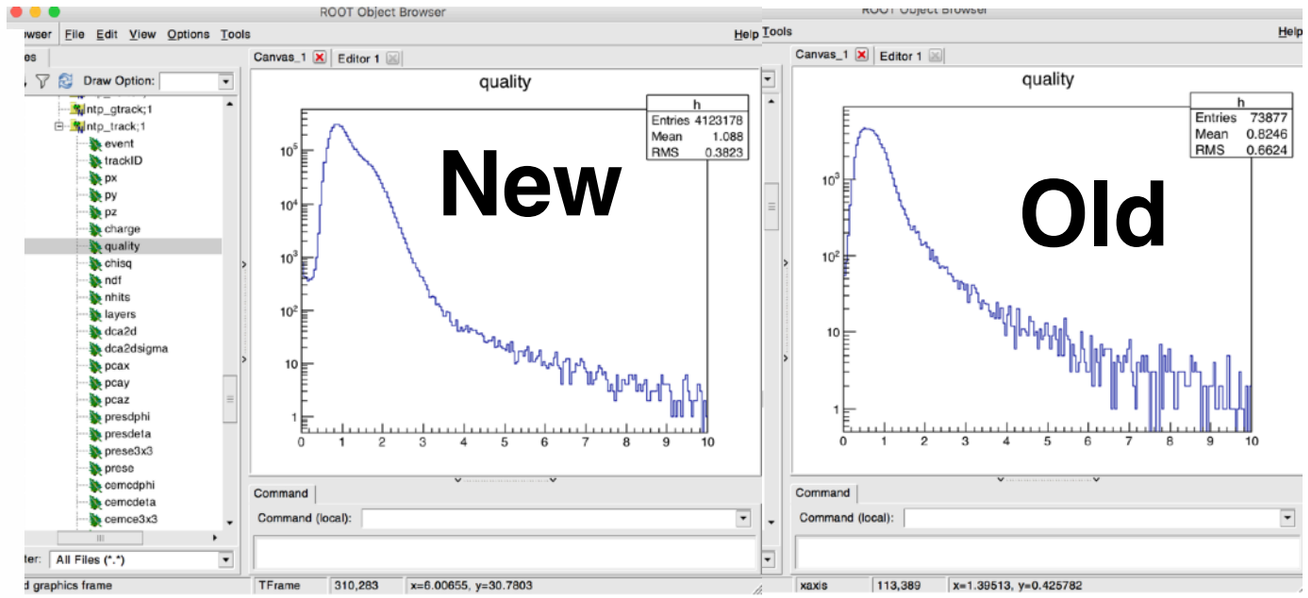
Track eff. is much lower

Why track eff. lower ?

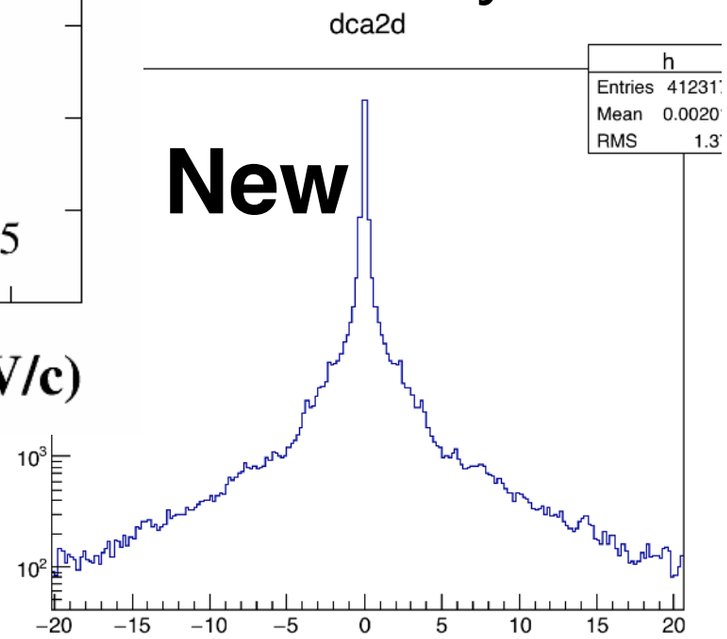
Single cut eff.

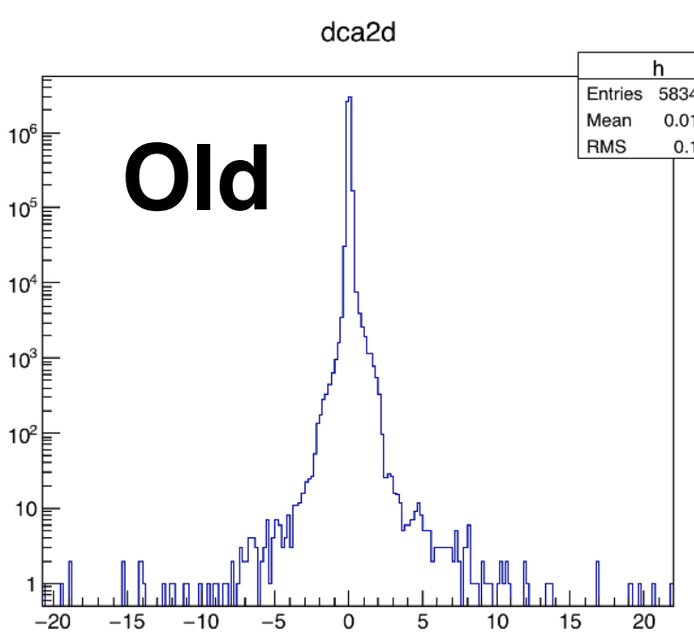


track quality



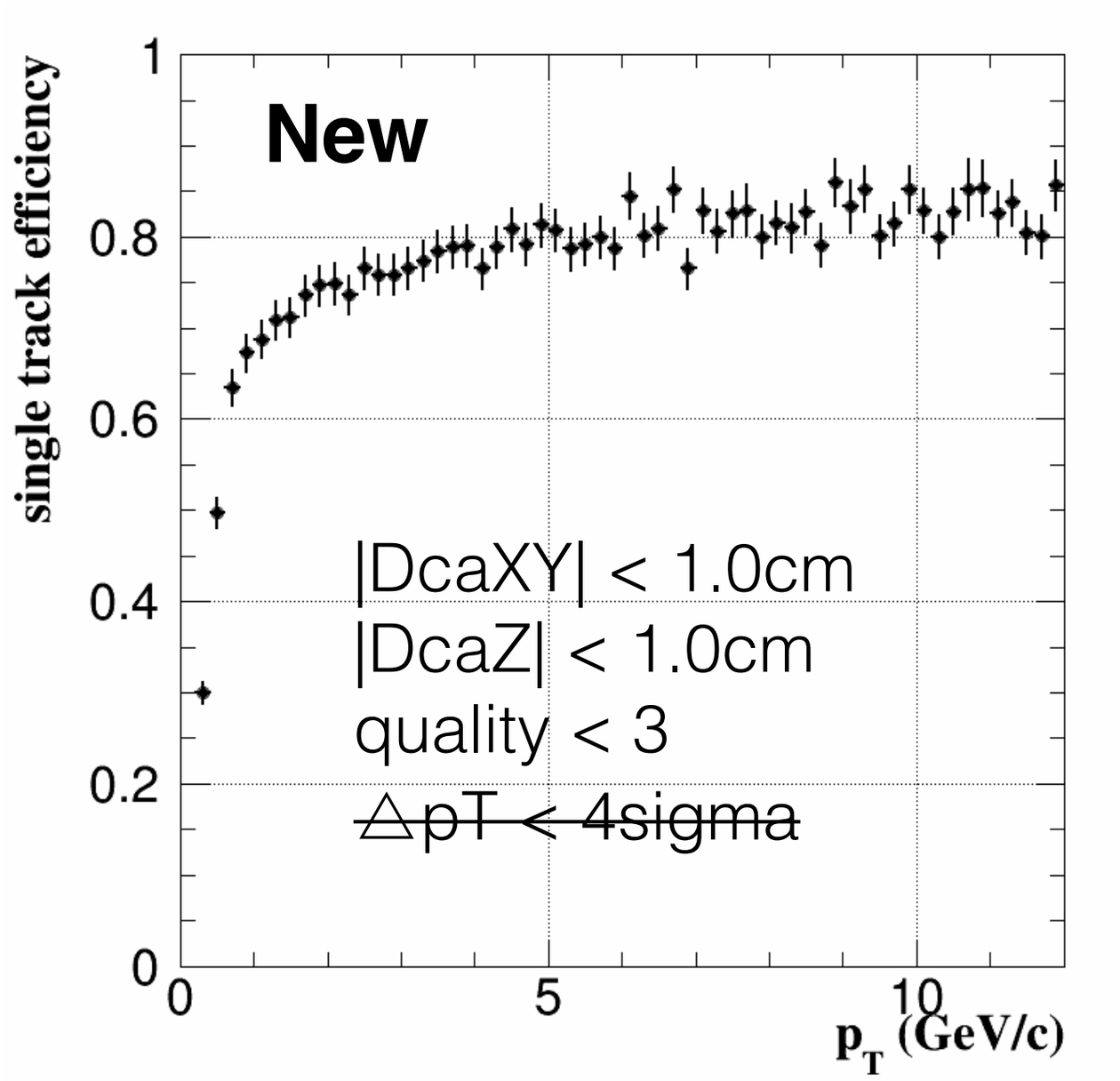
DCAxy

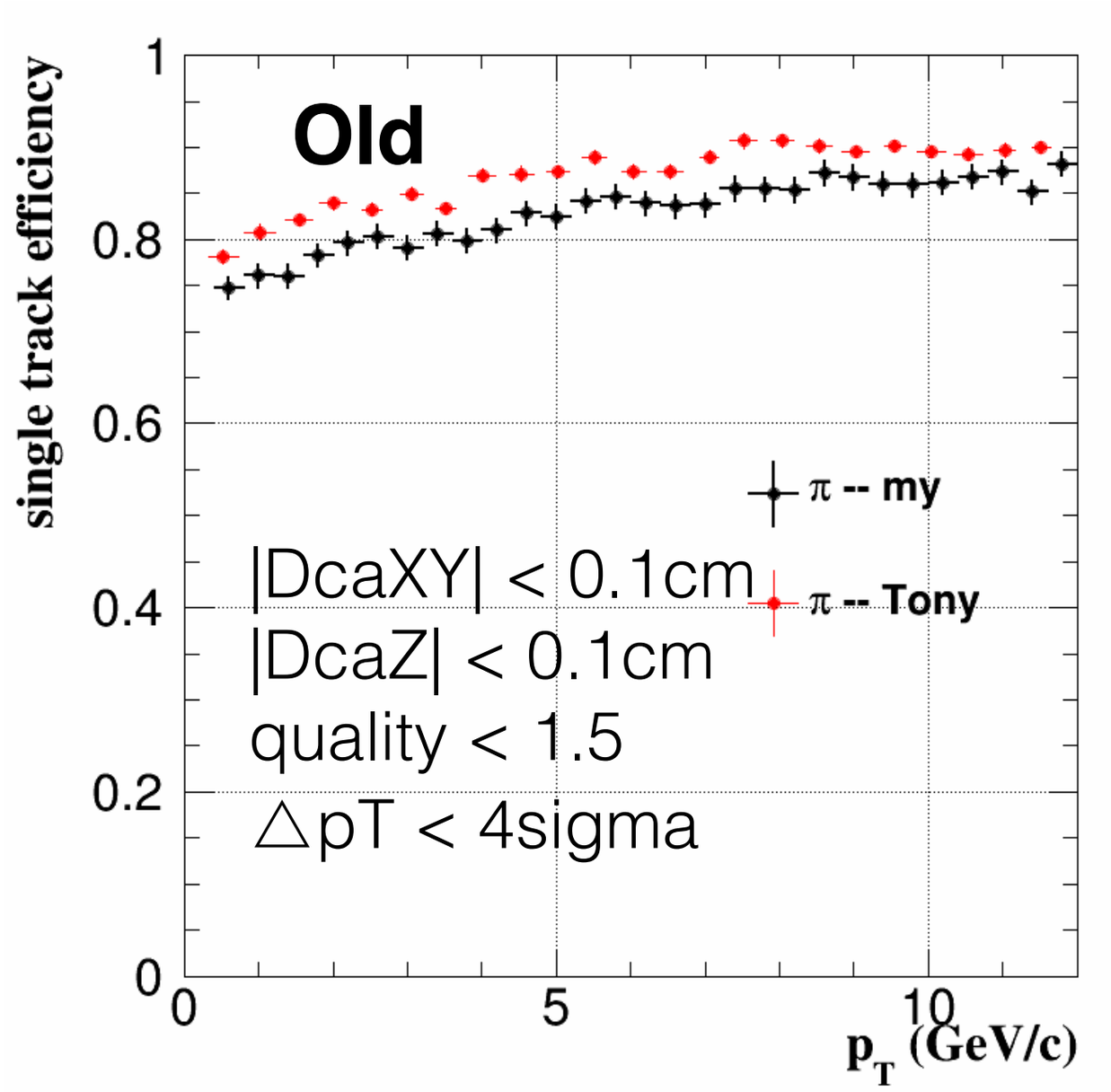




New production:
More wide track quality
More wide DCA distribution

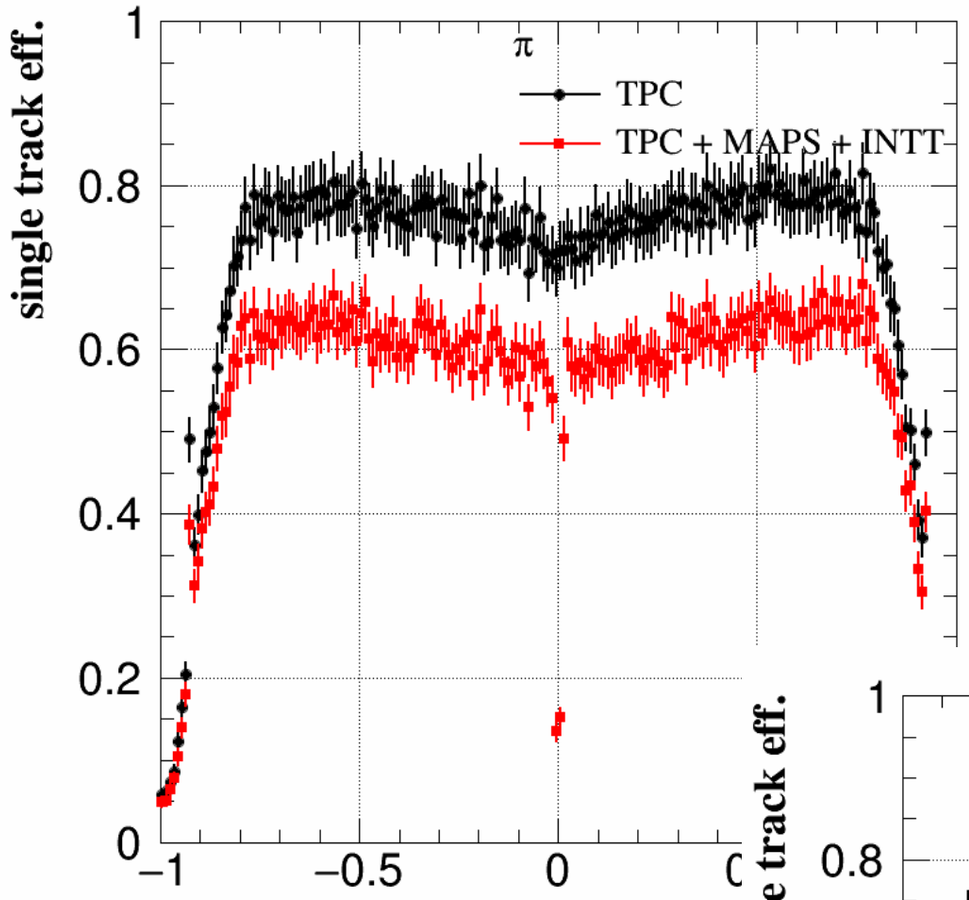
Try loose cut

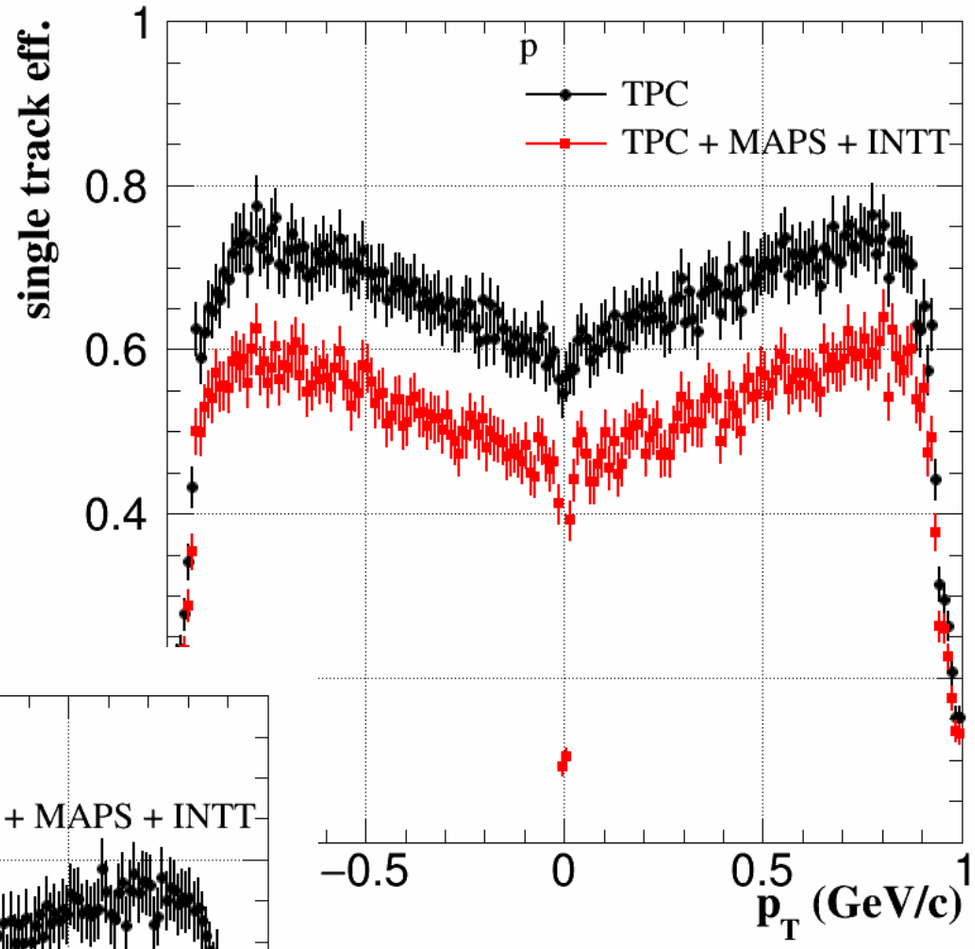


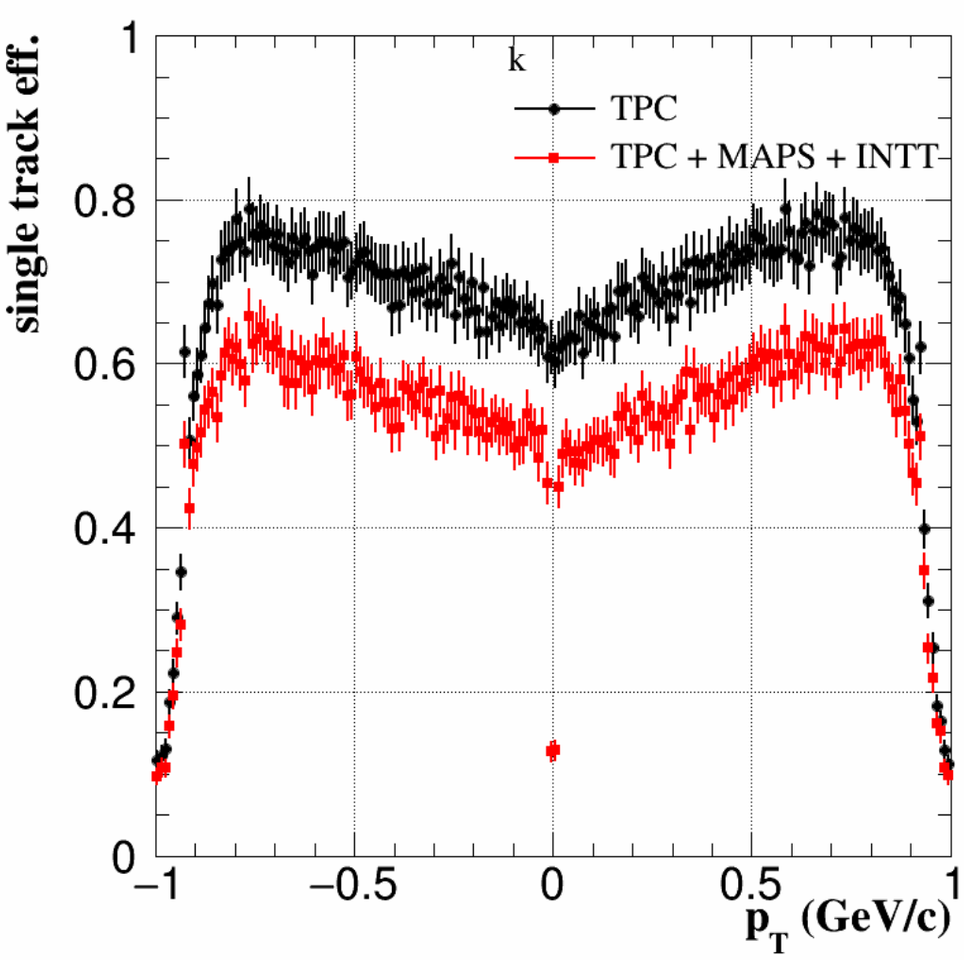


Get better, but still lower than old verision

Eff. in eta





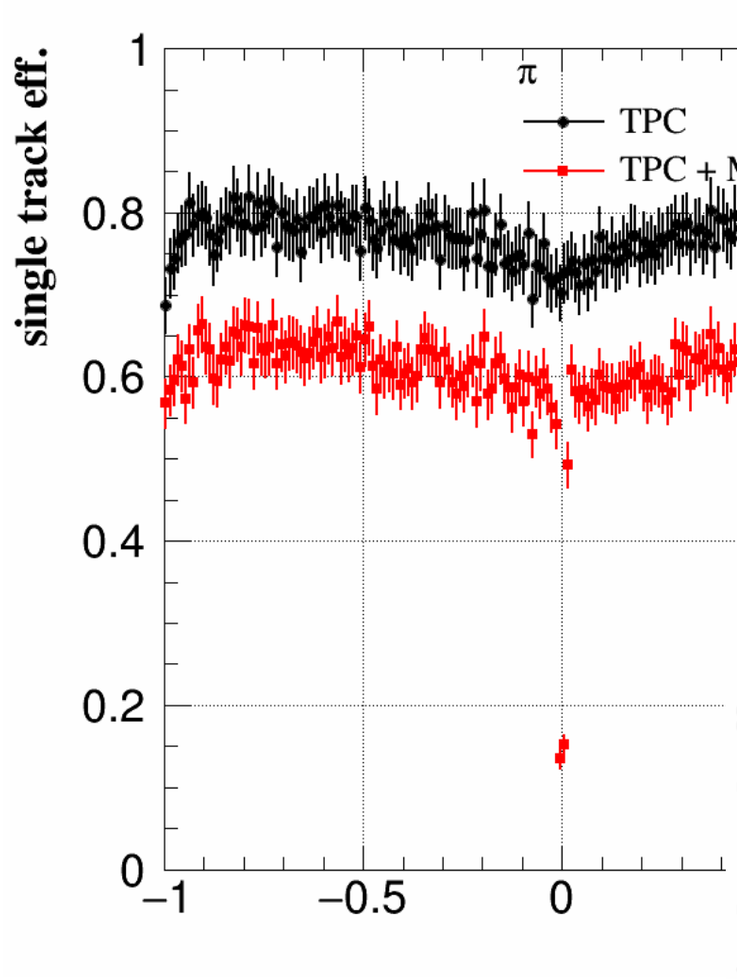


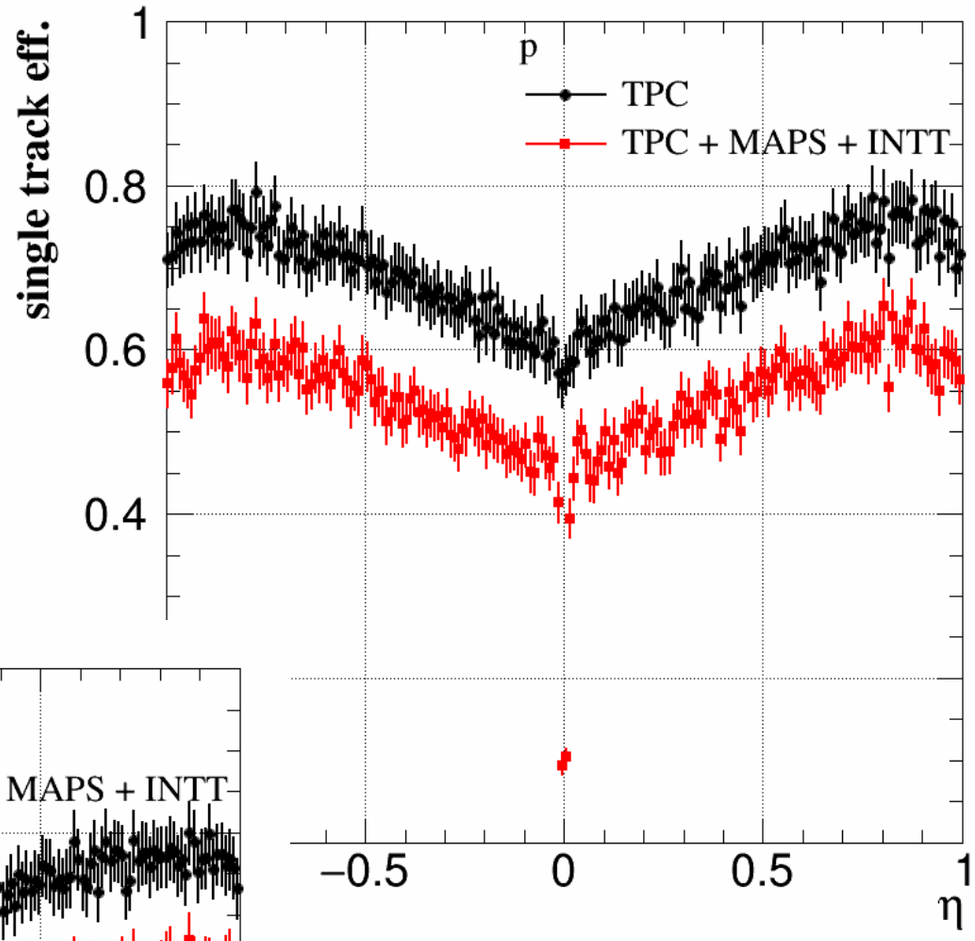
quality < 1

|DcaXY| < 1.0cm

|DcaZ| < 1.0cm

Eff. in eta



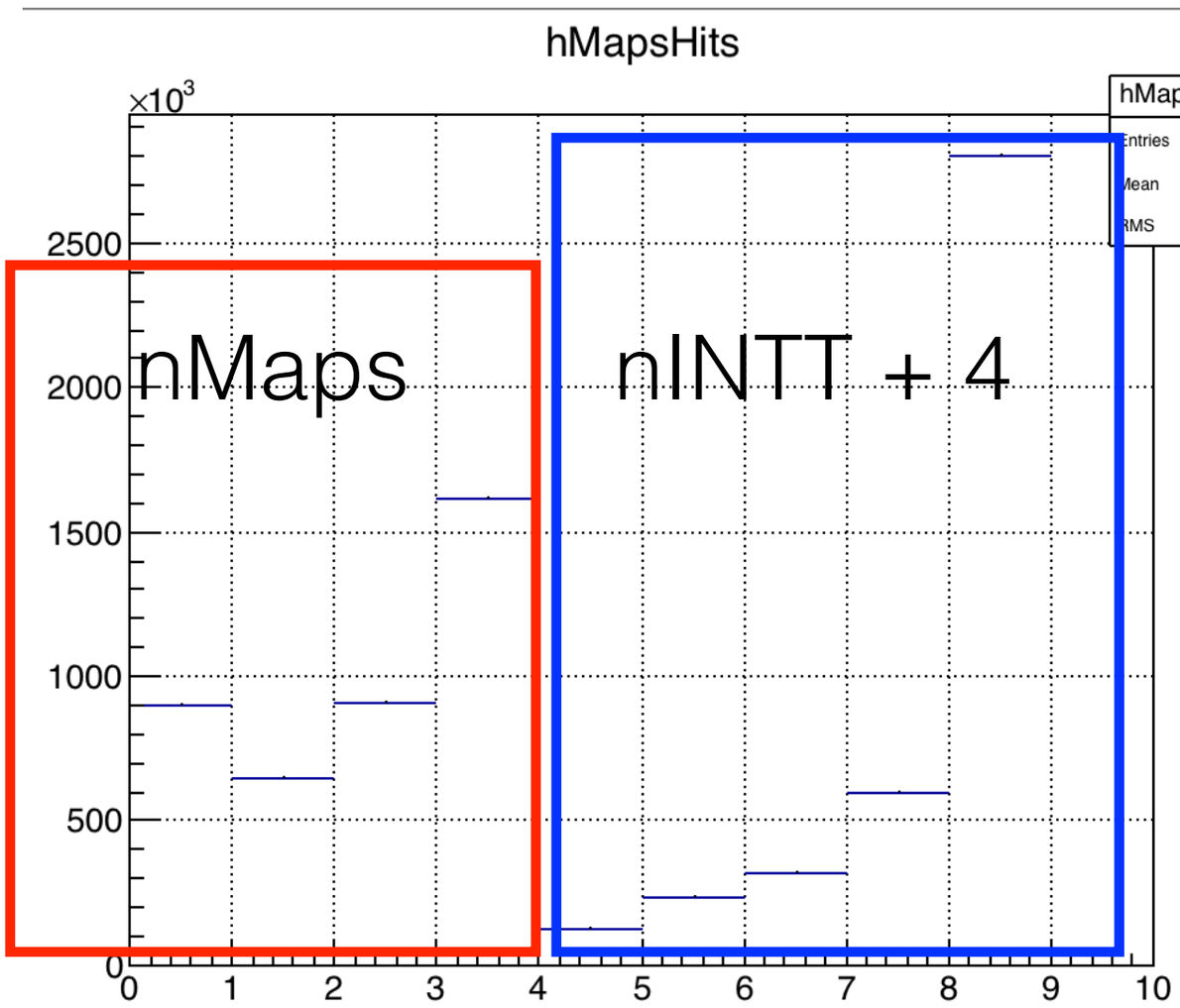


quality < 3

|DcaXY| < 1.0cm

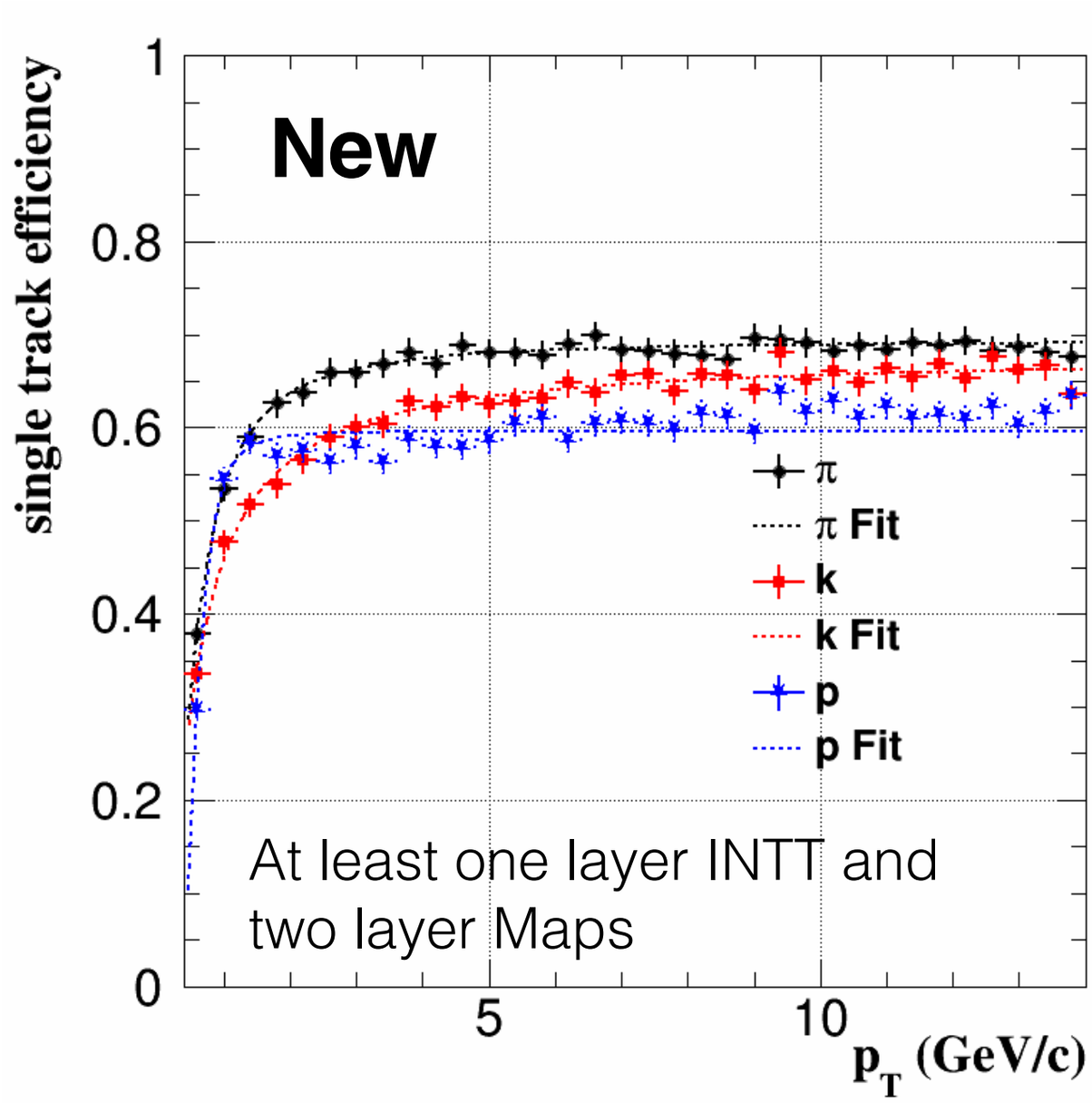
|DcaZ| < 1.0cm

Maps and INTT match

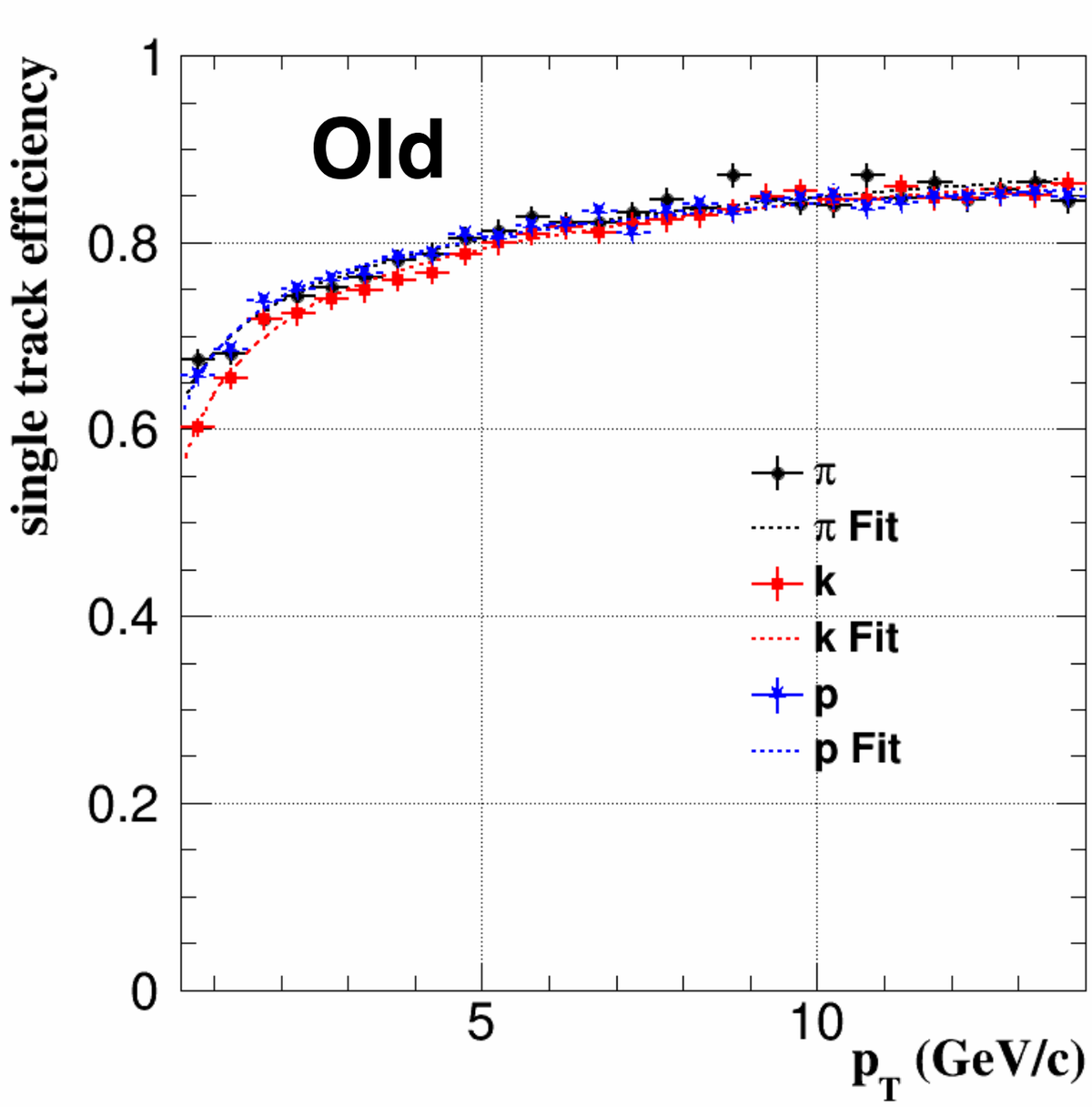



```
// maps layer number and INTT layer number
unsigned int layers = (floor)(tTrack->layers+0.5);
int nMaps = 0;
int nINTT = 0;
for(int i=0; i<7; i++) {
    bool isLayer = (layers>>i & 0x1);
    if(isLayer && i<3) nMaps++;
    else if(isLayer && i<7) nINTT++;
}

//nINTT += 3;
hMapsHits->Fill(nMaps);
hMapsHits->Fill(nINTT+4);

//maps and INTT Real match
unsigned int layersTru = (floor)(tTrack->layersfromtruth+0.5);
bool isRealMatch = true;
for(int i=0; i<7; i++) {
    bool ilayer = (layers>>i & 0x1);
    if(ilayer) {
        bool ilayerTru = (layersTru>>i & 0x1);
        if(!ilayerTru) isRealMatch = false;
    }
}
```

Track eff. with maps and INTT match

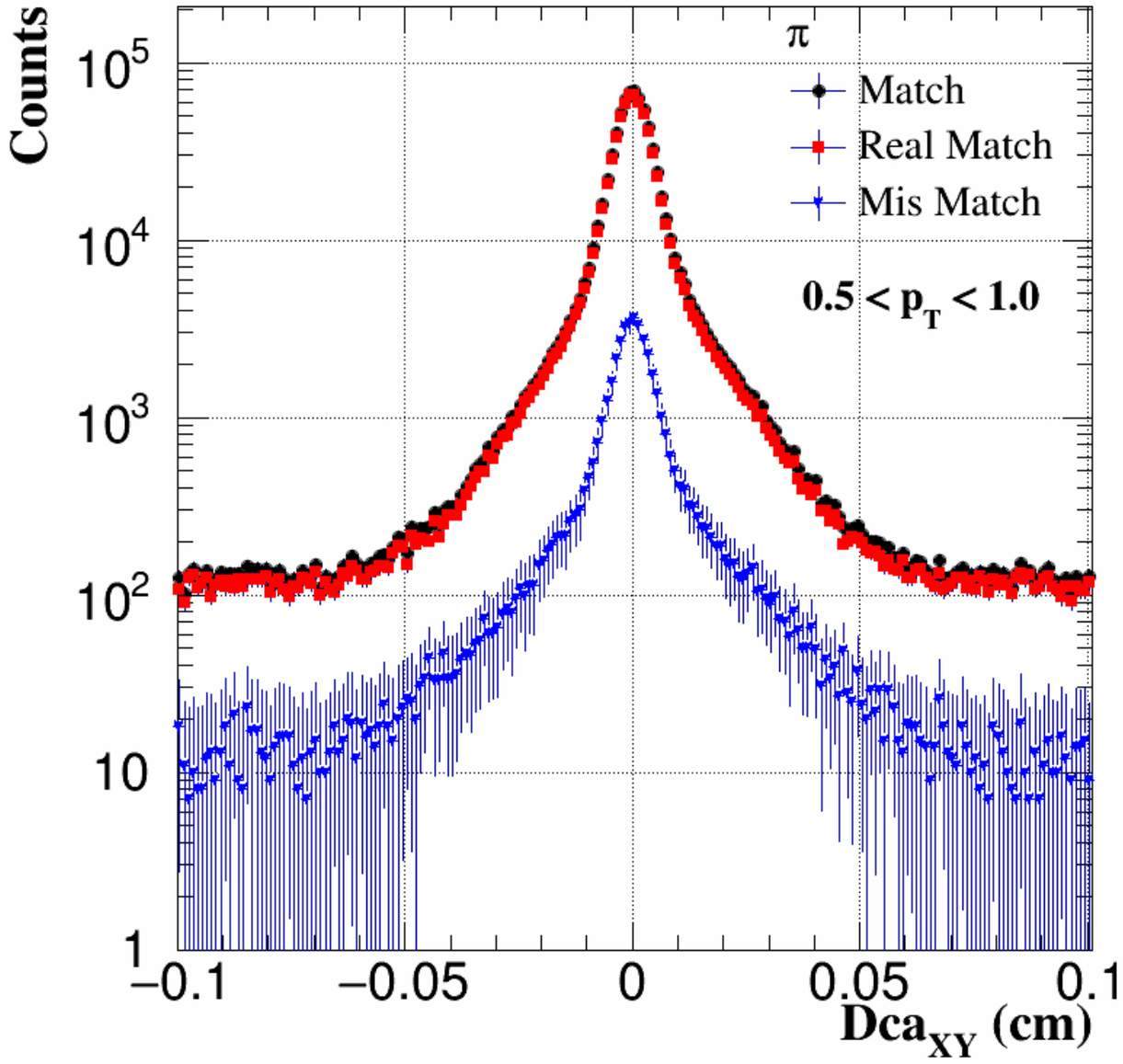


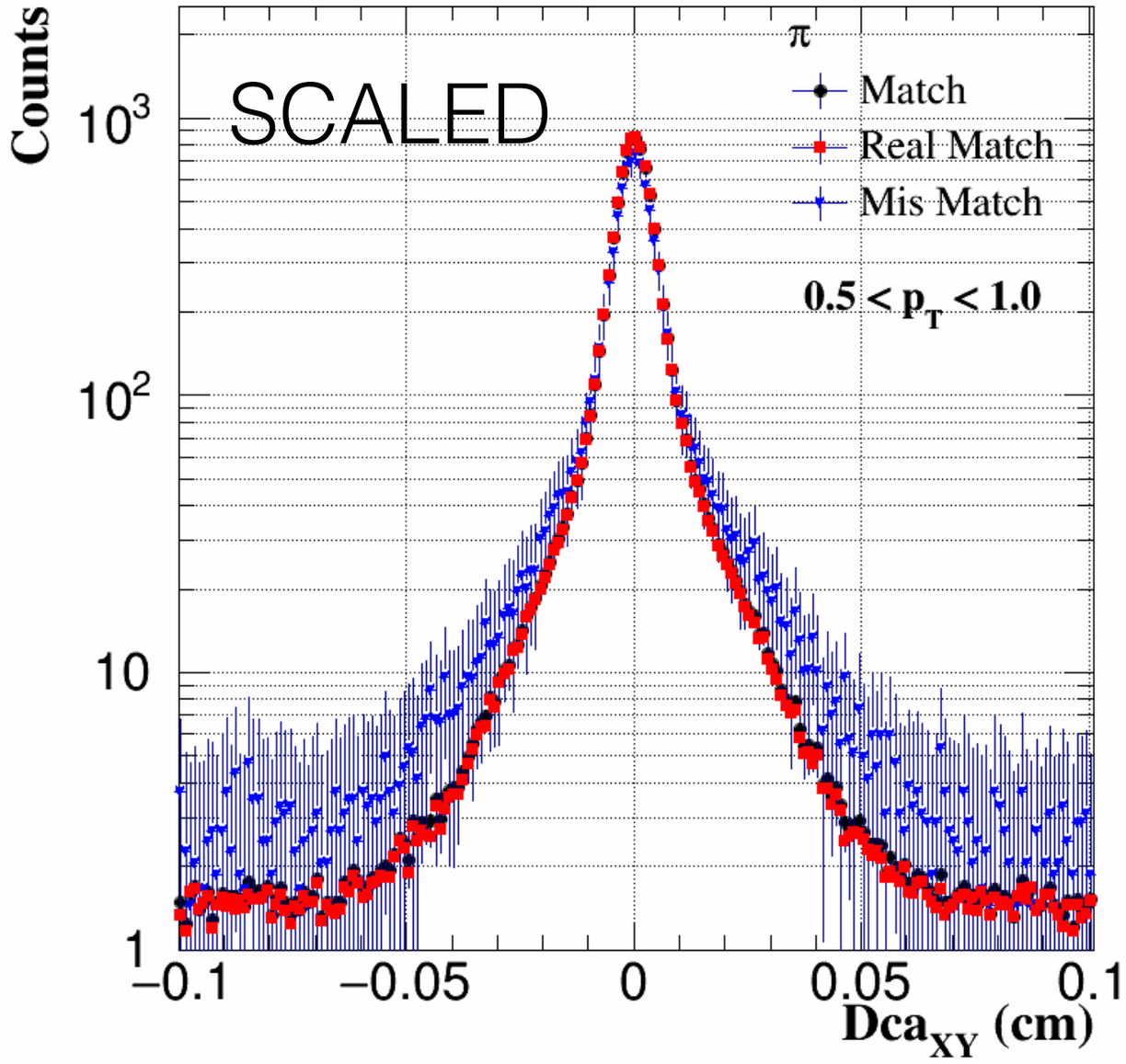
~~|DcaXY| < 1.0cm~~
~~|DcaZ| < 1.0cm~~
quality < 3



~~|DcaXY| < 0.1cm~~
~~|DcaZ| < 0.1cm~~
quality < 1.5

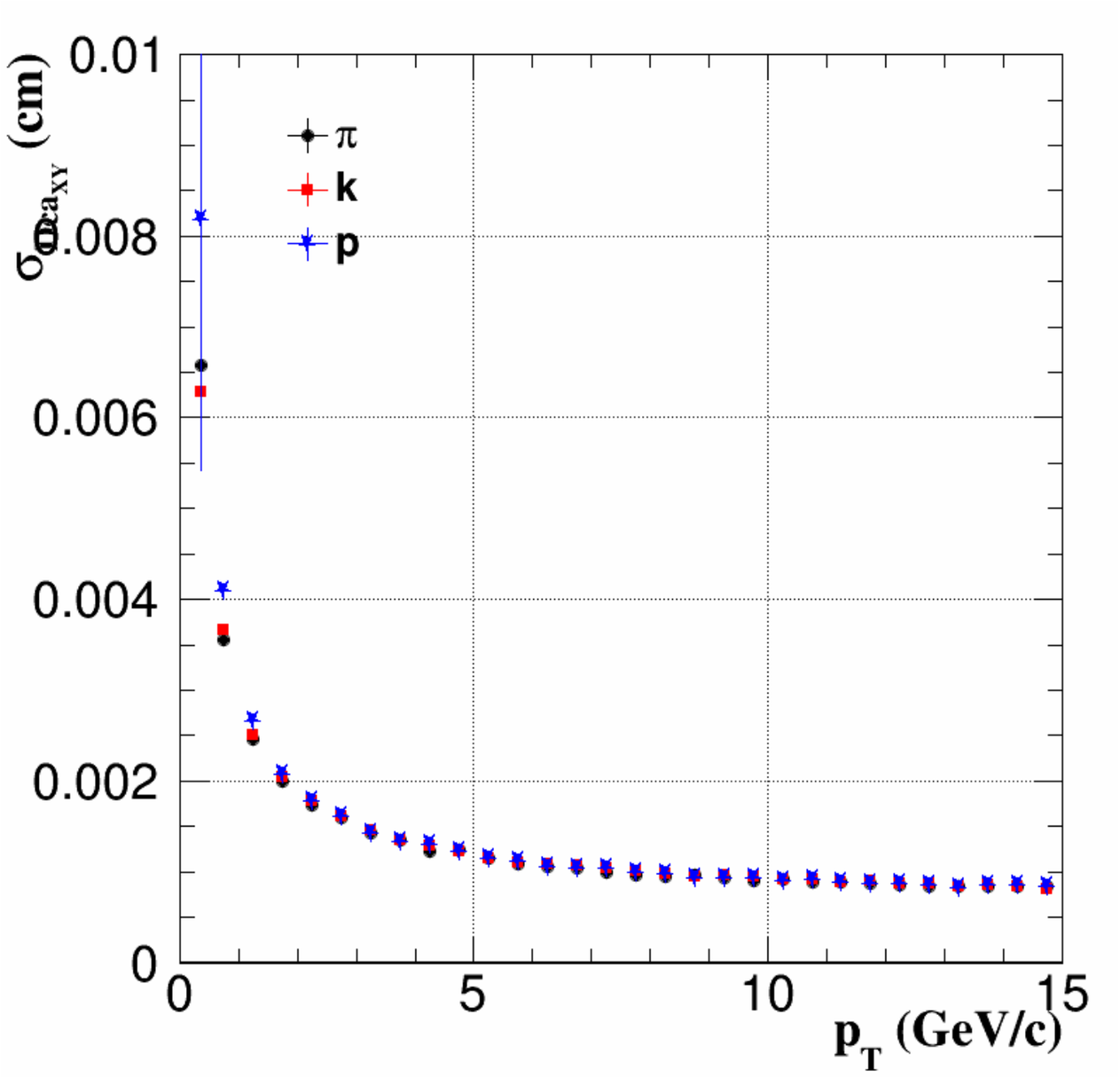
Maps/INTT mis-match

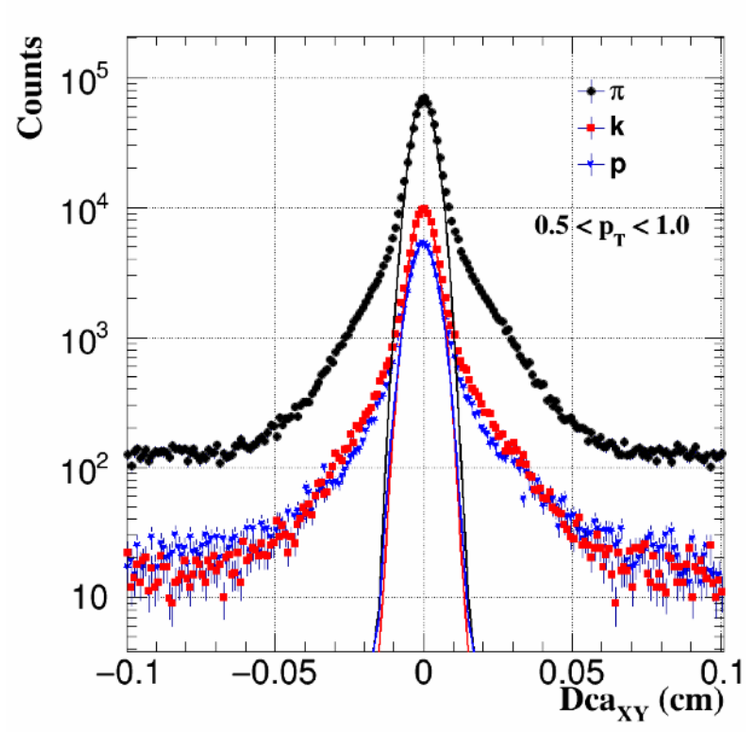


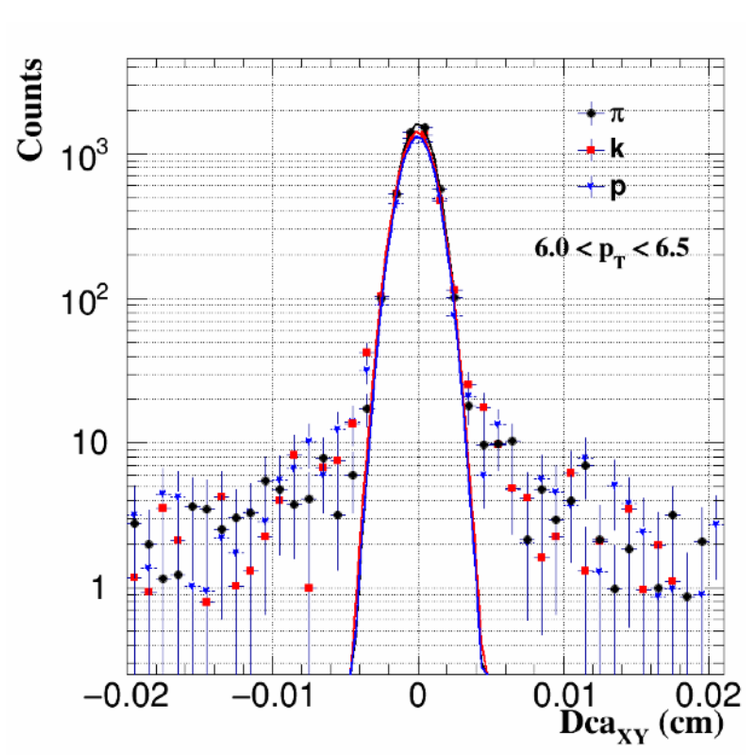


Mis-match track also have good DCA resolution ?
Long DCA tail is not only due to Maps/INTT mis-match
(Another reason for the large DCA tail is sometimes RAVE vertexing doesn't work)

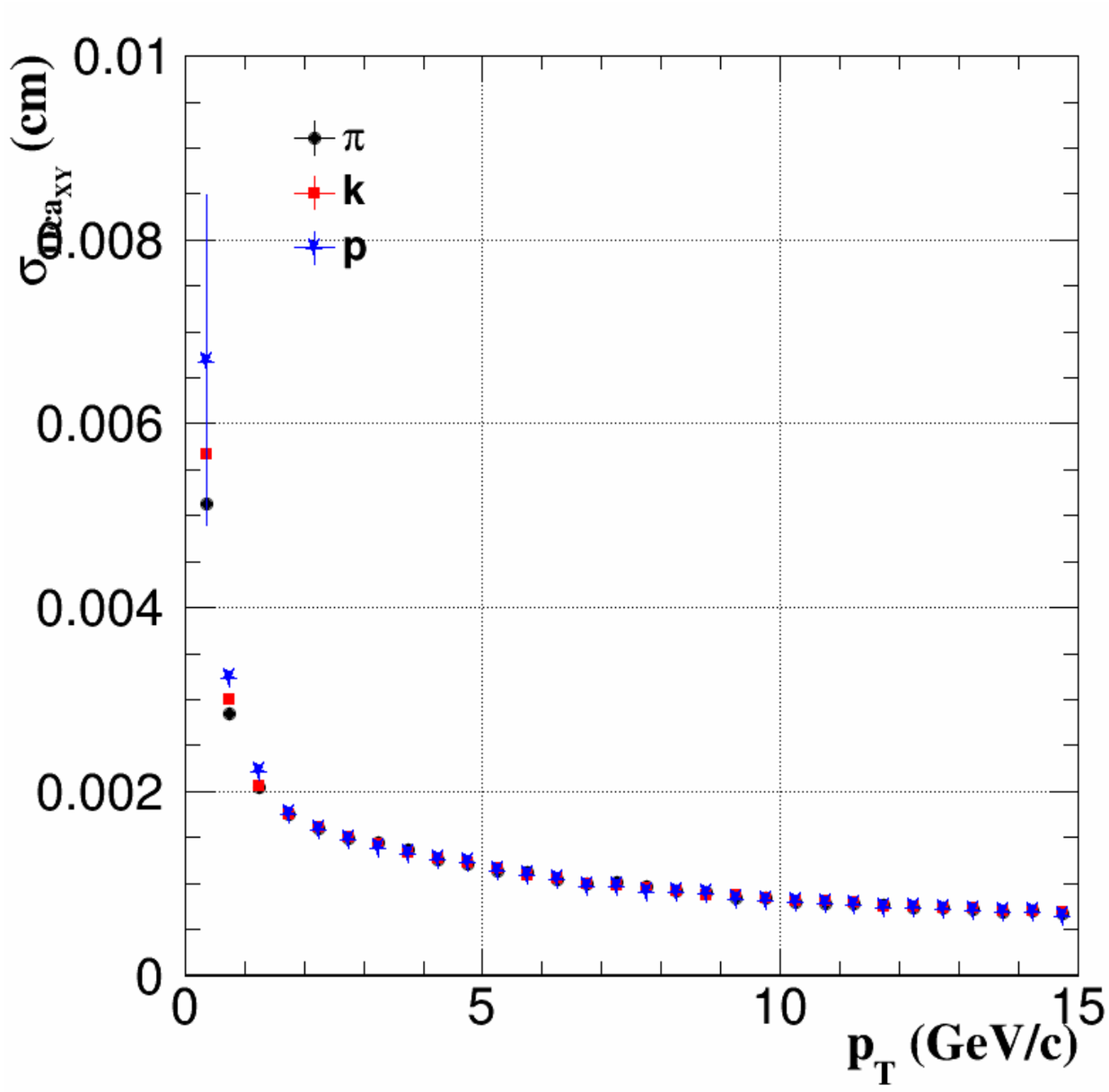
DCAxy reso.

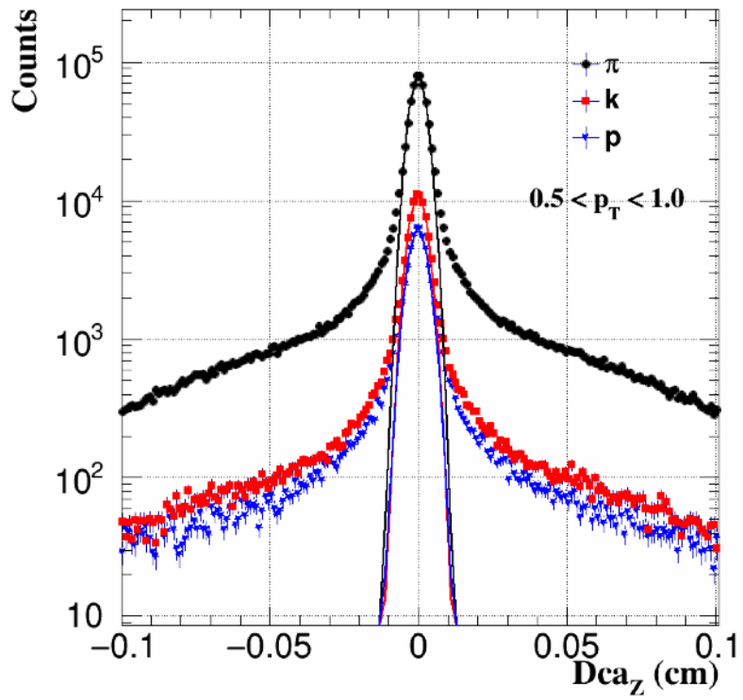

$\sigma_{Dca_{XY}}$ (cm)
0.01
0.008
0.006
0.004
0.002
0
π
k
p
0
5
10
15
p_T (GeV/c)

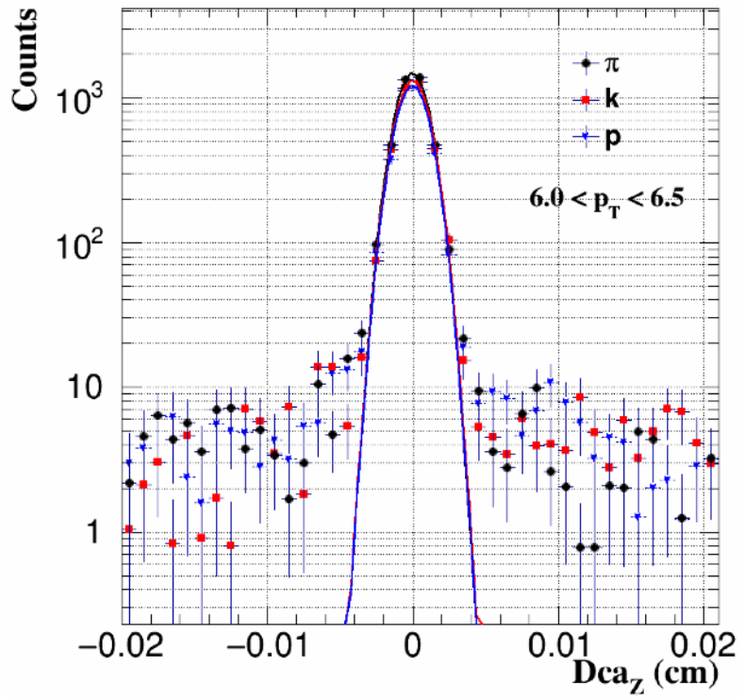

Counts
10^5
10^4
10^3
10^2
10
π
k
p
$0.5 < p_T < 1.0$
−0.1
−0.05
0
0.05
0.1
Dca_{XY} (cm)


Counts
10^3
10^2
10
1
π
k
p
$6.0 < p_T < 6.5$
−0.02
−0.01
0
0.01
0.02
Dca_{XY} (cm)

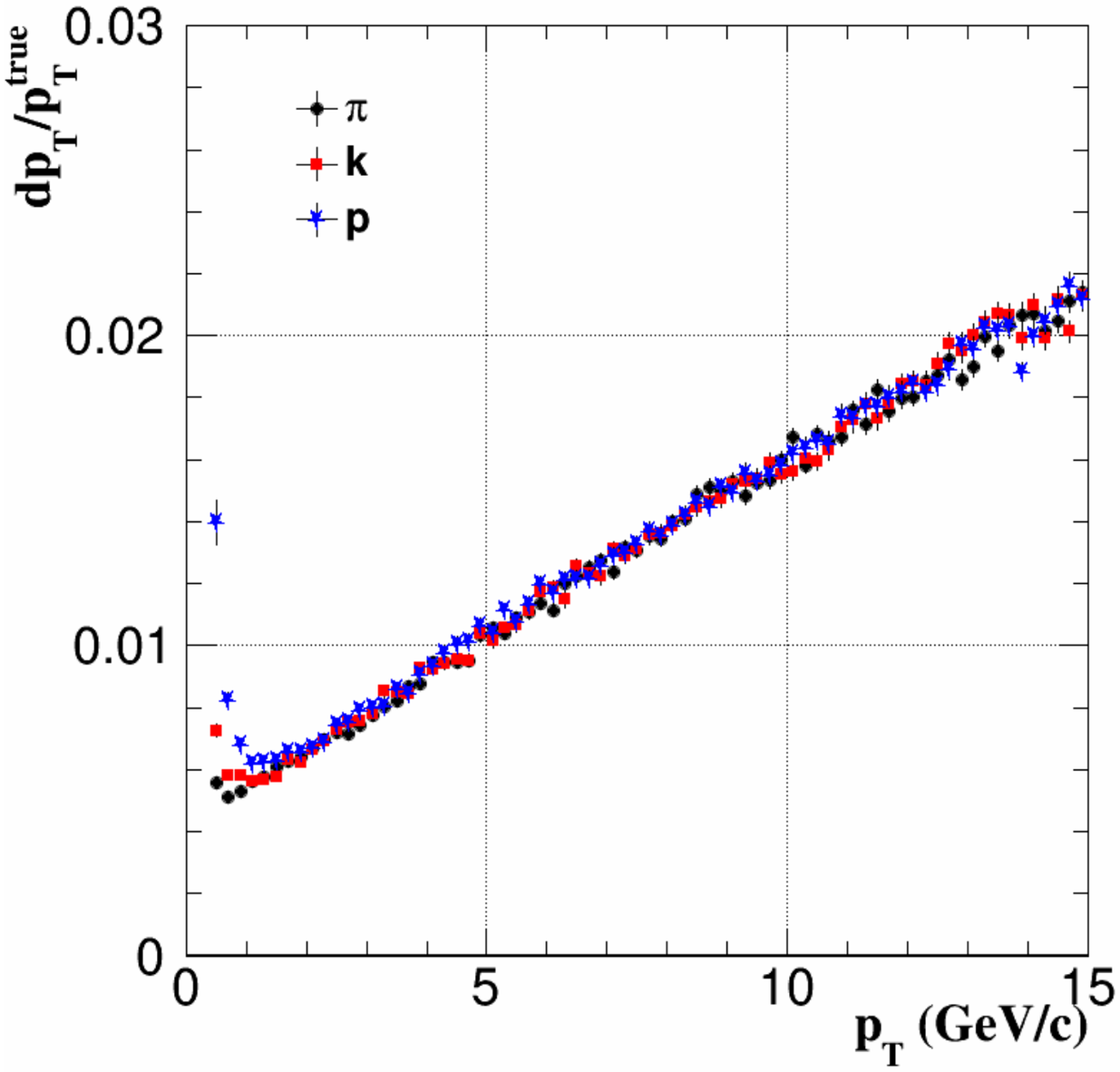
DCAz reso.

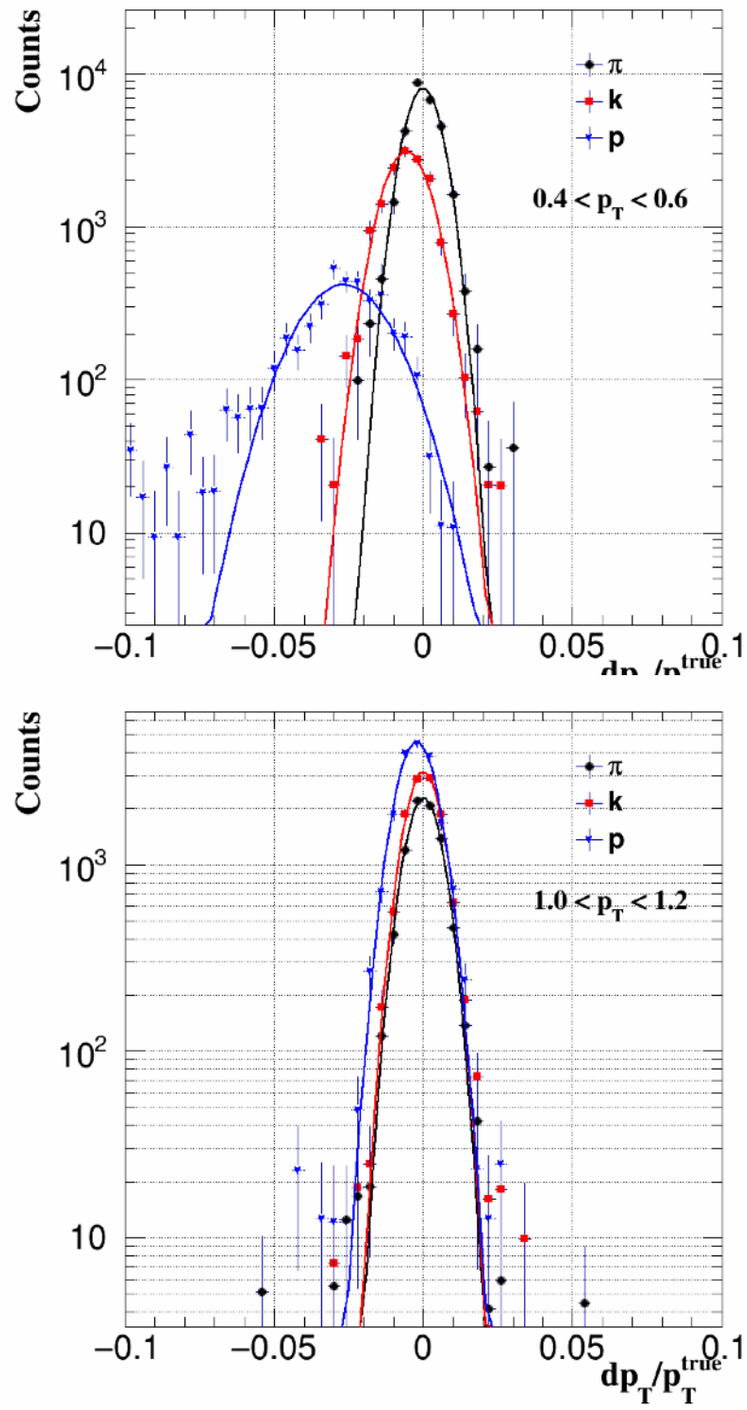

$\sigma_{Dca_{XY}}$ (cm)
0.01
0.008
0.006
0.004
0.002
0
π
k
p
0
5
10
15
p_T (GeV/c)


Counts
10^5
10^4
10^3
10^2
10
π
k
p
$0.5 < p_T < 1.0$
−0.1
−0.05
0
0.05
0.1
Dca_Z (cm)


Counts
10^3
10^2
10
1
π
k
p
$6.0 < p_T < 6.5$
−0.02
−0.01
0
0.01
0.02
Dca_Z (cm)

Momentum reso.





Proton mean value shifts at low Pt. (The Kalman filter used a pion-PID assumption in fit. It may not fit proton and Kaon well)

Some change in g4eval from github

TNtuple can’t save 32 bits, exceed the precise range

```
1. vary ntp_gtrack/layers 32(default) to 22
g4eval/SvtxEvaluator.C
line 1375: if (layer < 32) layers |= (0x1 << layer);
       ==> if (layer < 22) layers |= (0x1 << layer);

2. vary ntp_track/layers 31(default) to 22
g4eval/SvtxEvaluator.C
line 1477: if (layer < 31) layers |= (0x1 << layer);
       ==> if (layer < 22) layers |= (0x1 << layer);

3. vary ntp_gtrack/layersfromtruth and ntp_track/layersfromtruth from 30 layers (3FFFFFFF) to 22 layers (3FFFFF)
g4eval/SvtxTrackEval.C
line 493:  nclusters_by_layer |= (0x3FFFFFFF & (0x1 << cluster_layer));
       ==> if(cluster_layer < 22) nclusters_by_layer |= (0x3FFFFF & (0x1 << cluster_layer));
                                                                  needed, or layersfromtruth is not avaliable
4. add vtx information in ntp_track -- see g4eval_v3/SvtxEvaluator.C
```